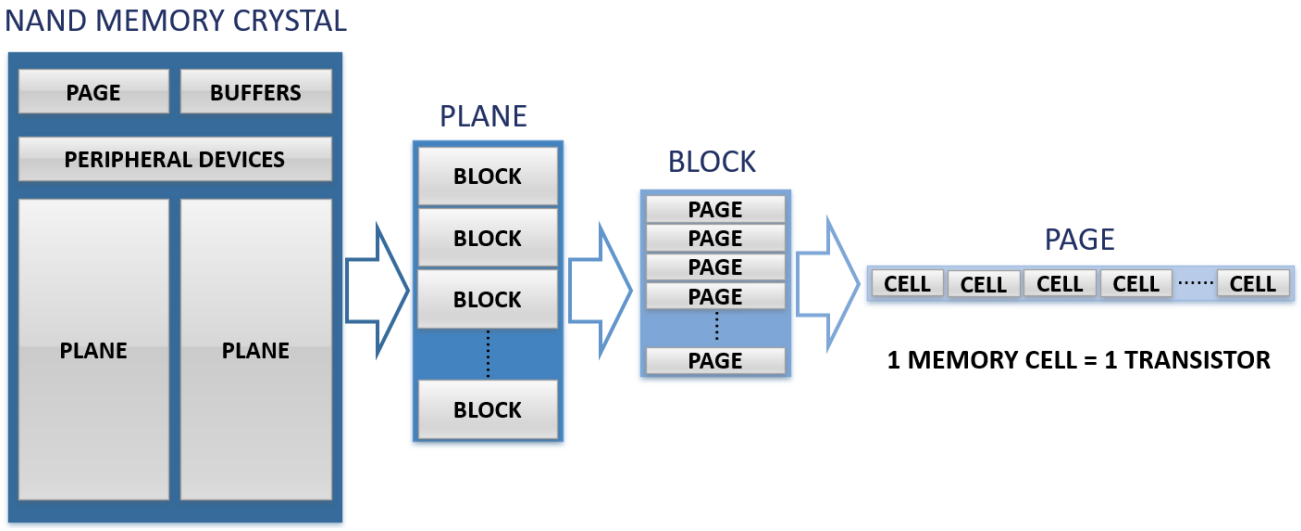
Help Center

Multi Plane Page Allocation

The base element of a NAND memory chip is a crystal and each crystal consist of several physical components: there are **peripheral devices** which provide I/O operation, page decoding and overall communication between NAND crystal and memory controller. There are **page buffers** for temporal data-keeping. And there are **planes** where user data is written and stored permanently. Single NAND crystal consists of one, two or four such planes in a majority of cases .



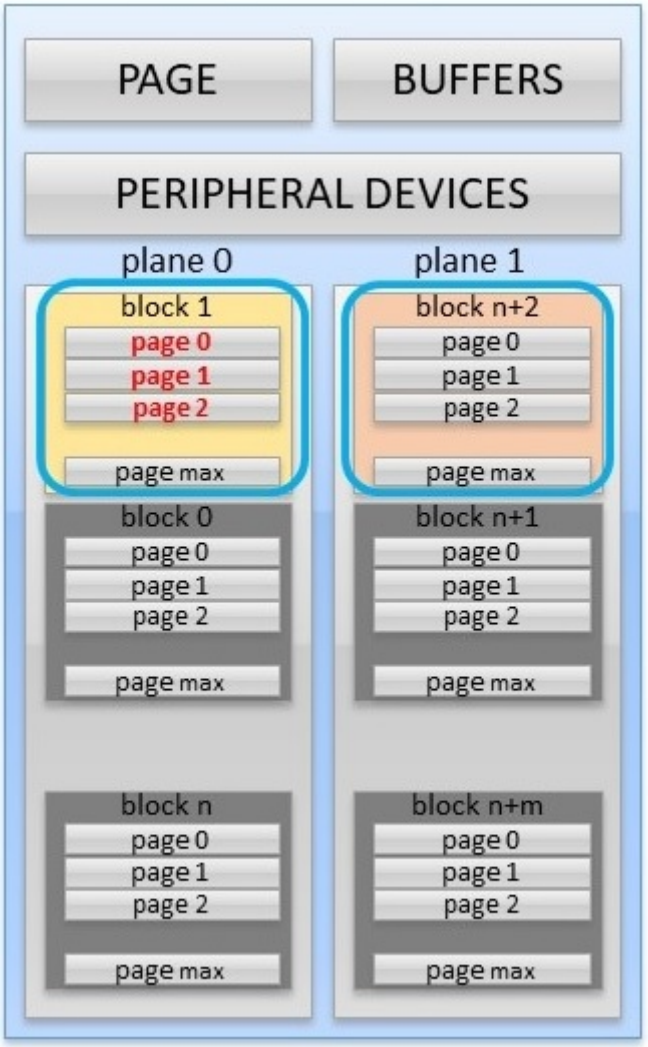
The controller writes data by pages. It can write them in two ways to those planes:

Single plane page allocation - sequential page writing to one physical blocks of one plane
Multiplane page allocation - parallel page writing into the physical blocks of each plane (RAID 0 like)

Single plane page allocation

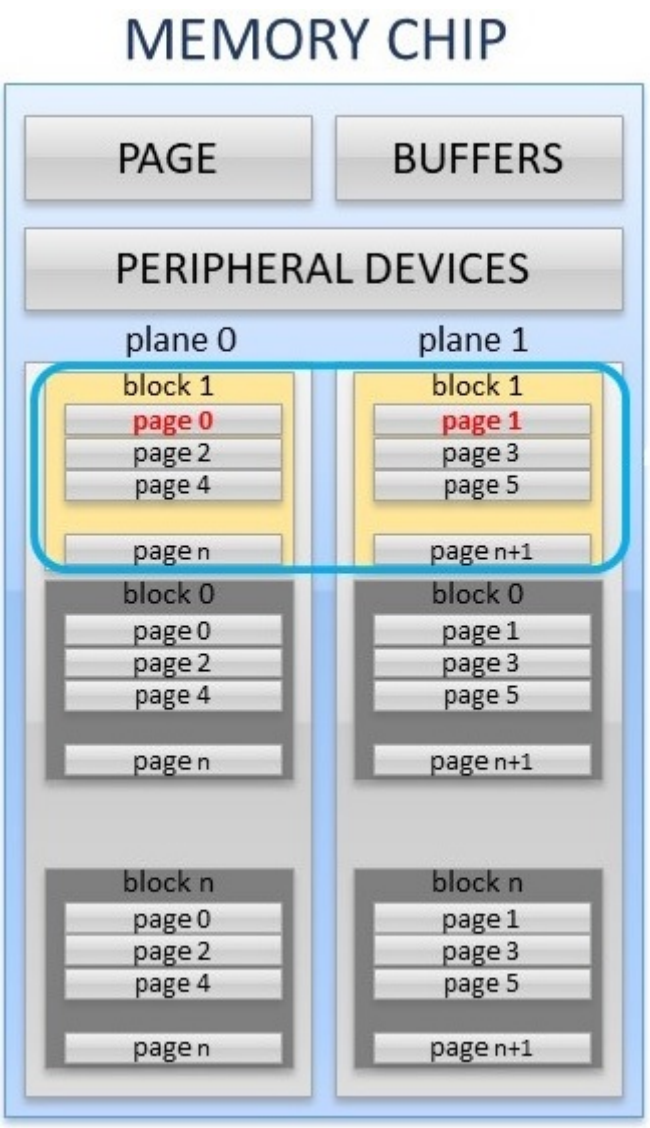
Memory controller that uses single plane page allocation mode,

allocate pages of data sequentially to one physical blocks.

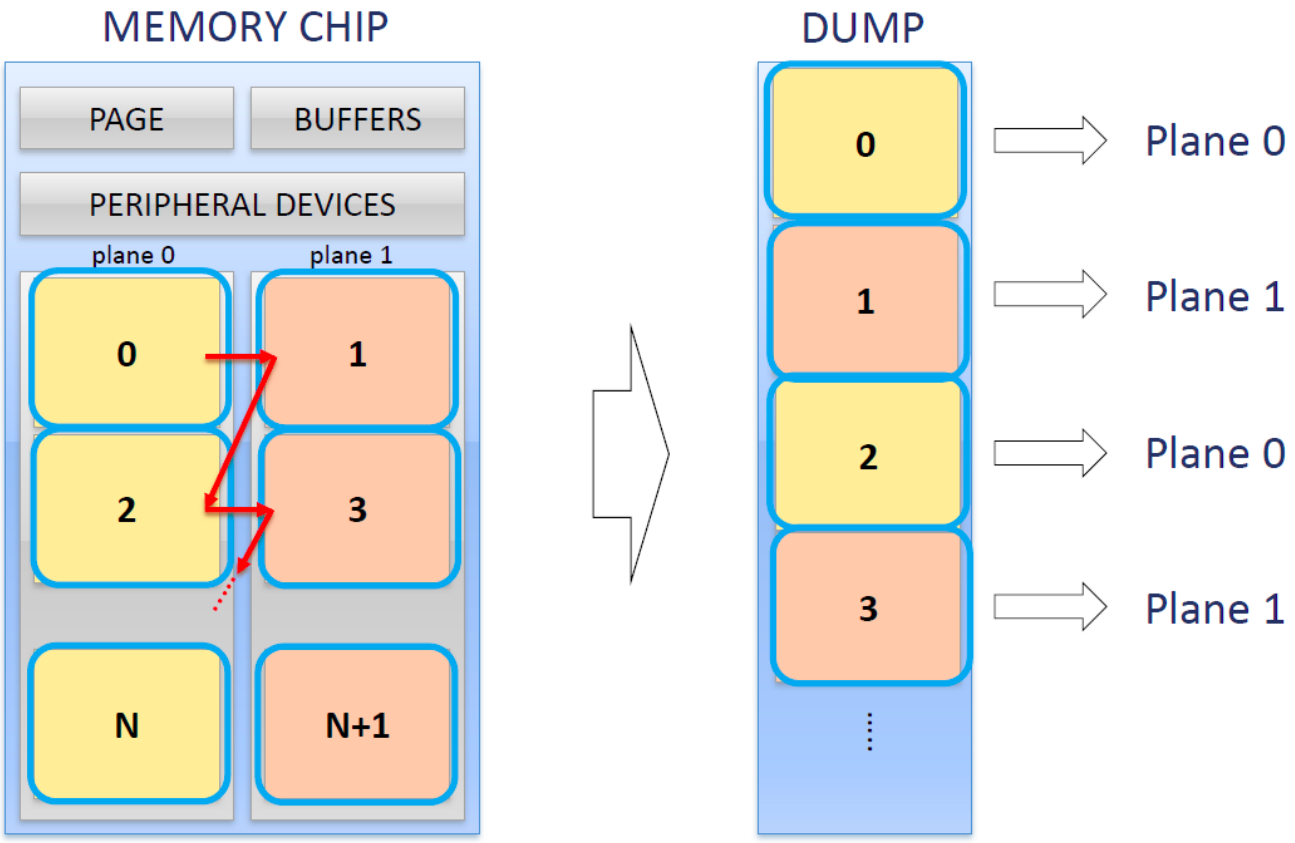


Multiplane page allocation

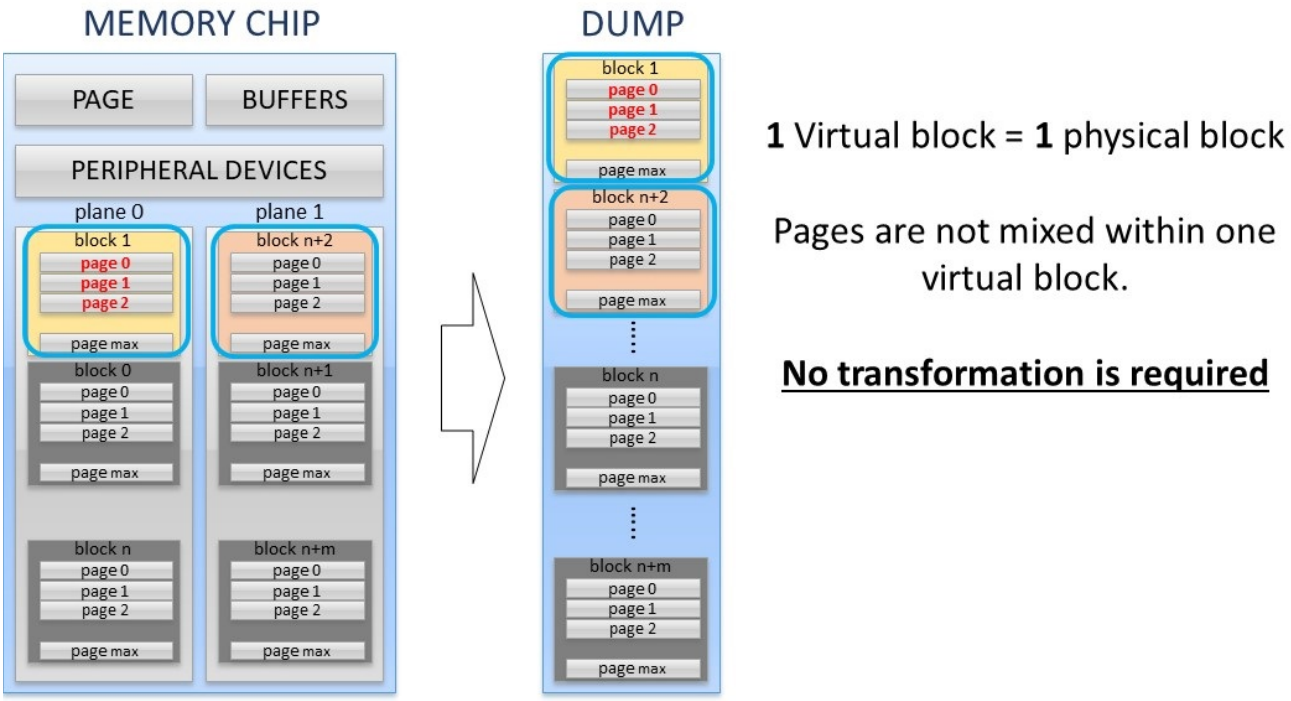
When a device is using Multiplane page allocation mode, pages of data are recorded simultaneously to the physical blocks of each plane. The reason why this mode is commonly used is the much higher speed of I/O operations than Single Plane page allocation.



During physical image extraction, data from memory crystals are read by whole blocks from every plane. It means that at first, a whole physical block from the first plane is read and in the second cycle, a whole physical block from the second plane is read.

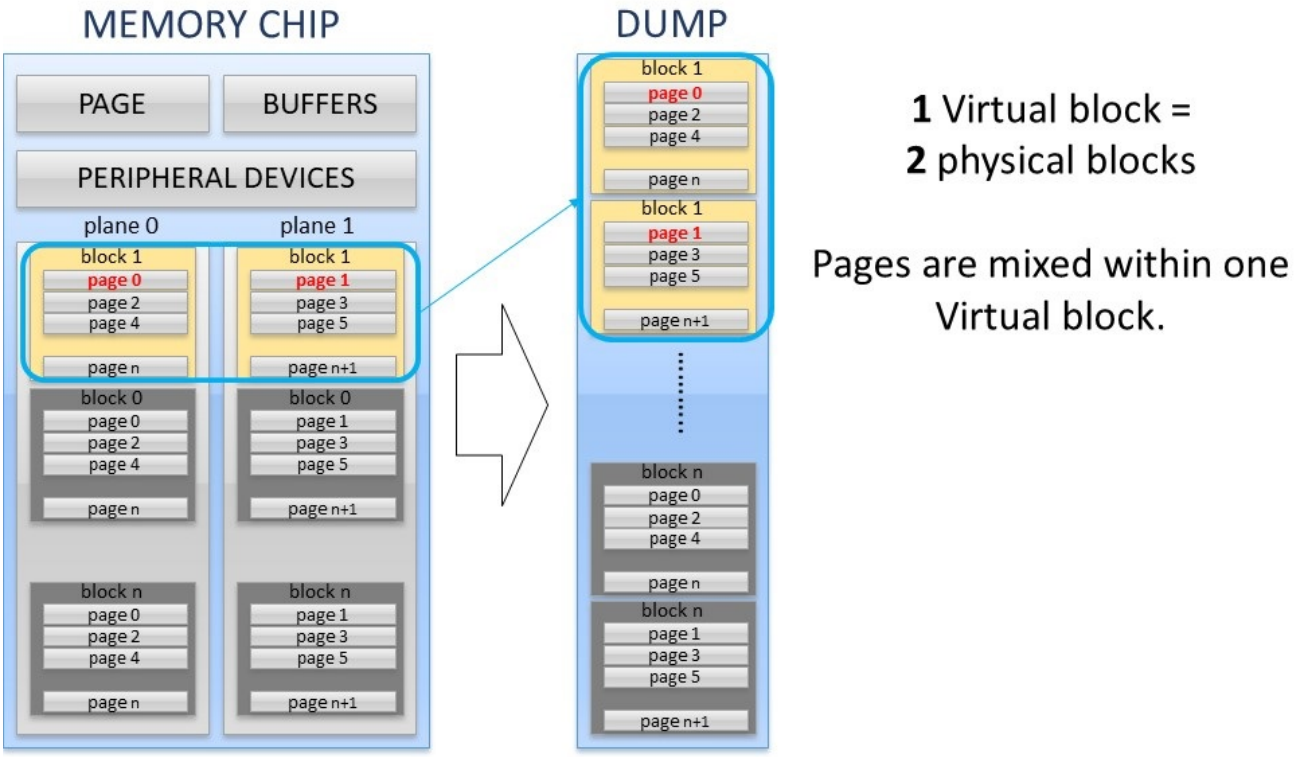


In case of Single plane page allocation, pages are deployed in proper order within one physical blocks. So when blocks are read to a dump, pages are not mixed there, one virtual block has a size of one physical block.

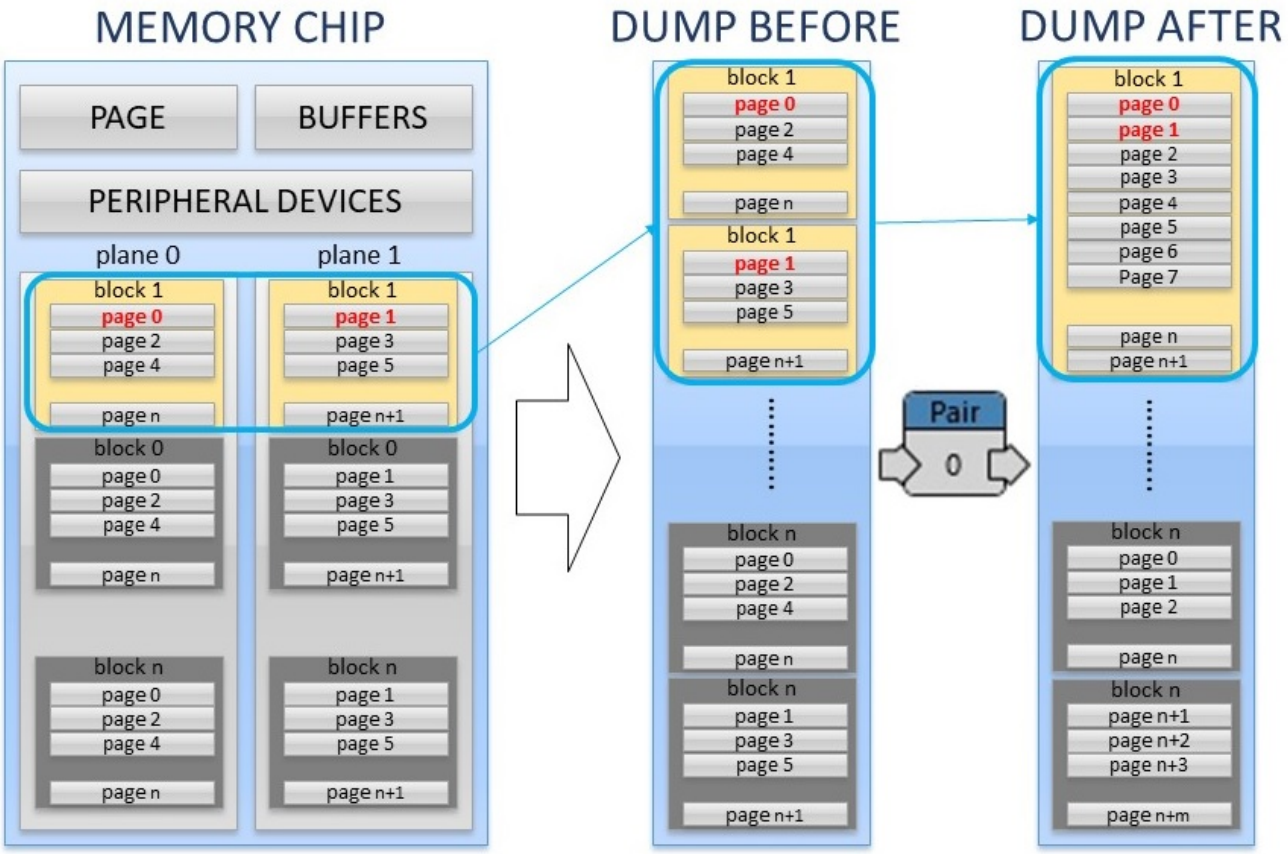


When the controller uses Multiplane page allocation, physical blocks that belong to the Plane 0 consist of 'even' pages and physical blocks of the Plane 1 have 'odd' pages. B locks from these planes are placed

one after another in the dump because VNR reads data by whole physical blocks. Therefore both these physical blocks will be on the neighbour positions in the dump. Both of them belong to one Virtual block.



As a result, pages of one Virtual block are mixed in the dump. In this case, it's necessary to put pages back on their logical positions and this operation can be performed with the **PAIR** element.



Pair element has a set of parameters that has to be determined:

- A number of planes (2/4/8/16...)
- Movable block size - defines page size
- Area size - virtual block size (physical block size * number of planes)

VNR has predefined Virtual block size which is equal to one Physical block usually, but in case of Multiplane page allocation, it may be necessary to adjust the size of Virtual block. Virtual block size determination and adjustment are described in the article Phison dynamic XOR

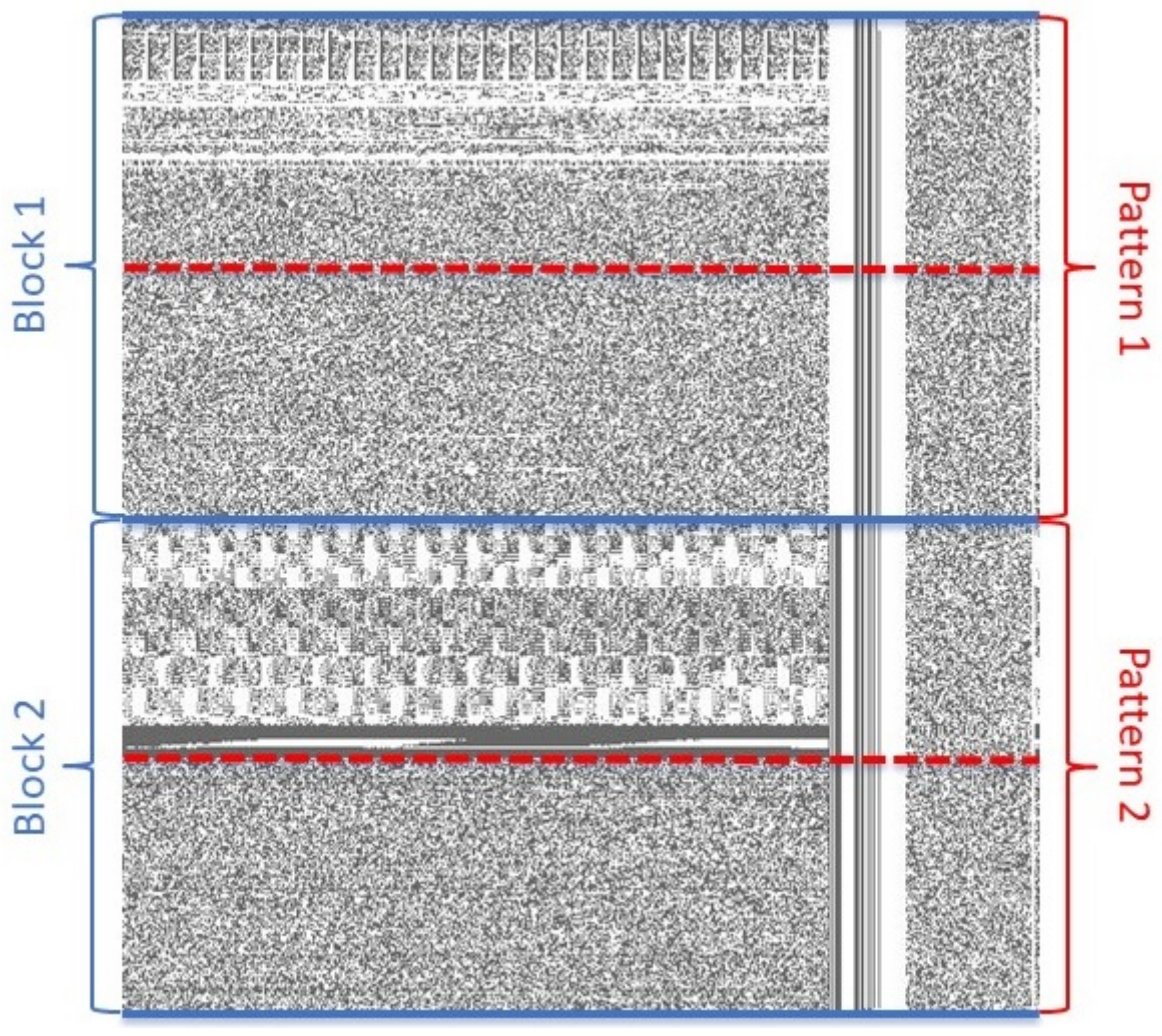
Page allocation scheme analysis

One of the methods to d istinguish whether the controller used MultiPlane Page allocation or Single Plane page allocation is data pattern analysis which can be done via Bitmap Viewer. In this method, it is important to find a proper block which has a characteristic pattern and then according to this pattern to detect which page allocation scheme device was using.

ⓘ Pattern analysis should be performed on unXORed data, otherwise, it will be impossible to notice patterns.

Single plane page allocation

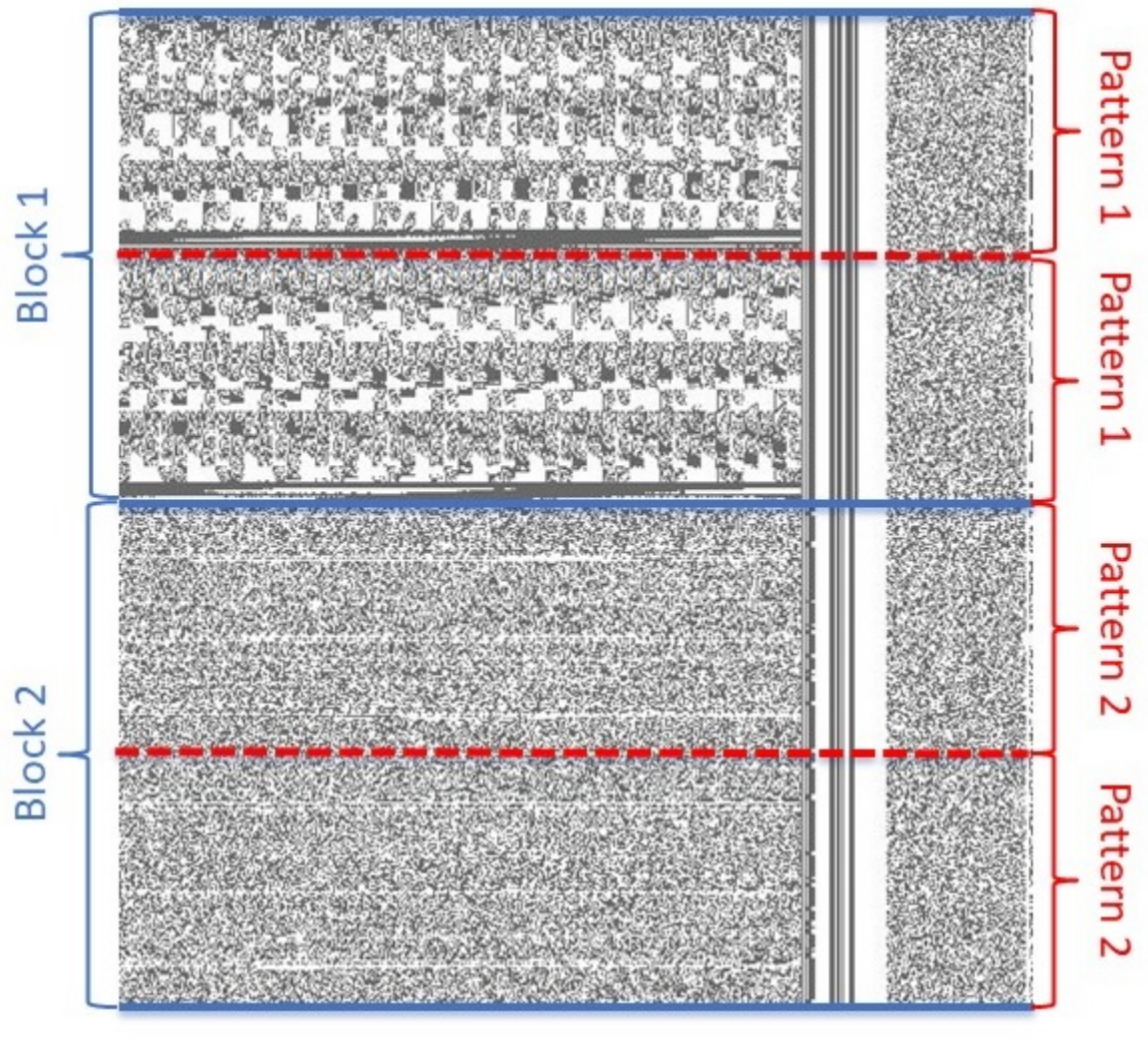
Single plane page allocation has a pattern of data, that was written sequentially, looks continuously from the beginning of the virtual block to the end of it, neighbour blocks have different patterns and LBN's. In this situation, it's not necessary to apply Pair element.



Multiplane page allocation

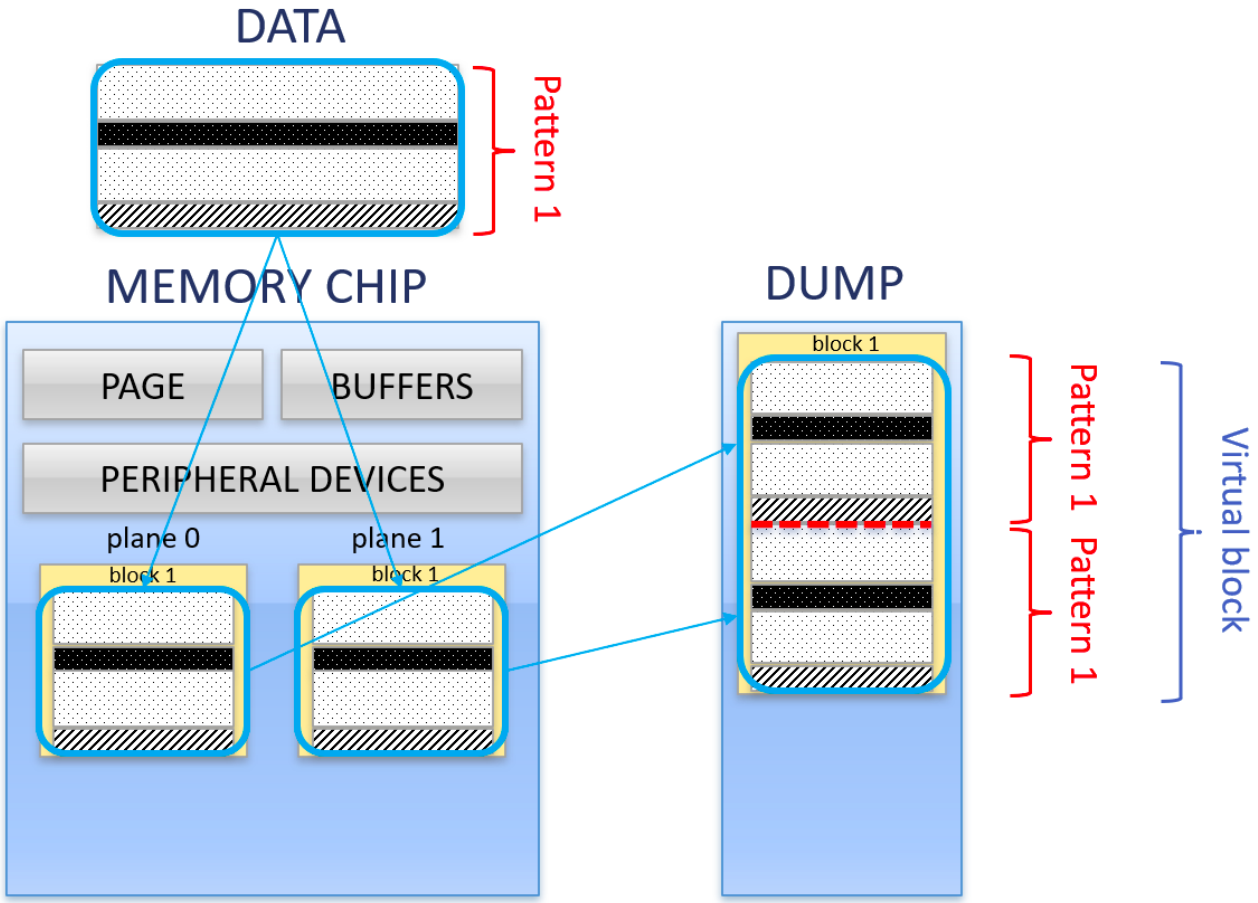
Data that was written page by page into two planes simultaneously have a similar pattern in two neighbour physical blocks and the same LBN. These two physical blocks form one Virtual block, so in other words, it can be described as - Multiplane page allocation is

characterized by a repeatable pattern of data within one Virtual block.



ⓘ In a situation when a crystal has 4 planes, there are 4 physical blocks with a similar pattern and the same LBN's within one Virtual block.

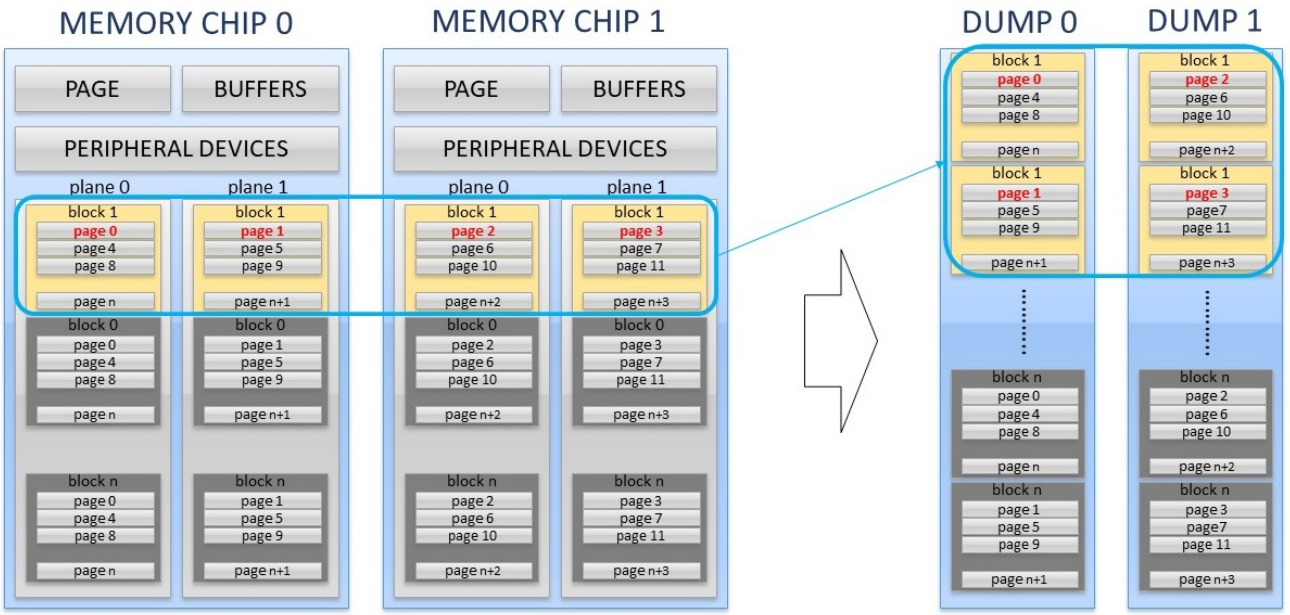
The duplicated pattern is the effect of parallel page allocation, it occurs because pages of one portion of data were recorded simultaneously into two planes, therefore two physical blocks have comparable patterns.



MultiPlane page allocation on the chip level

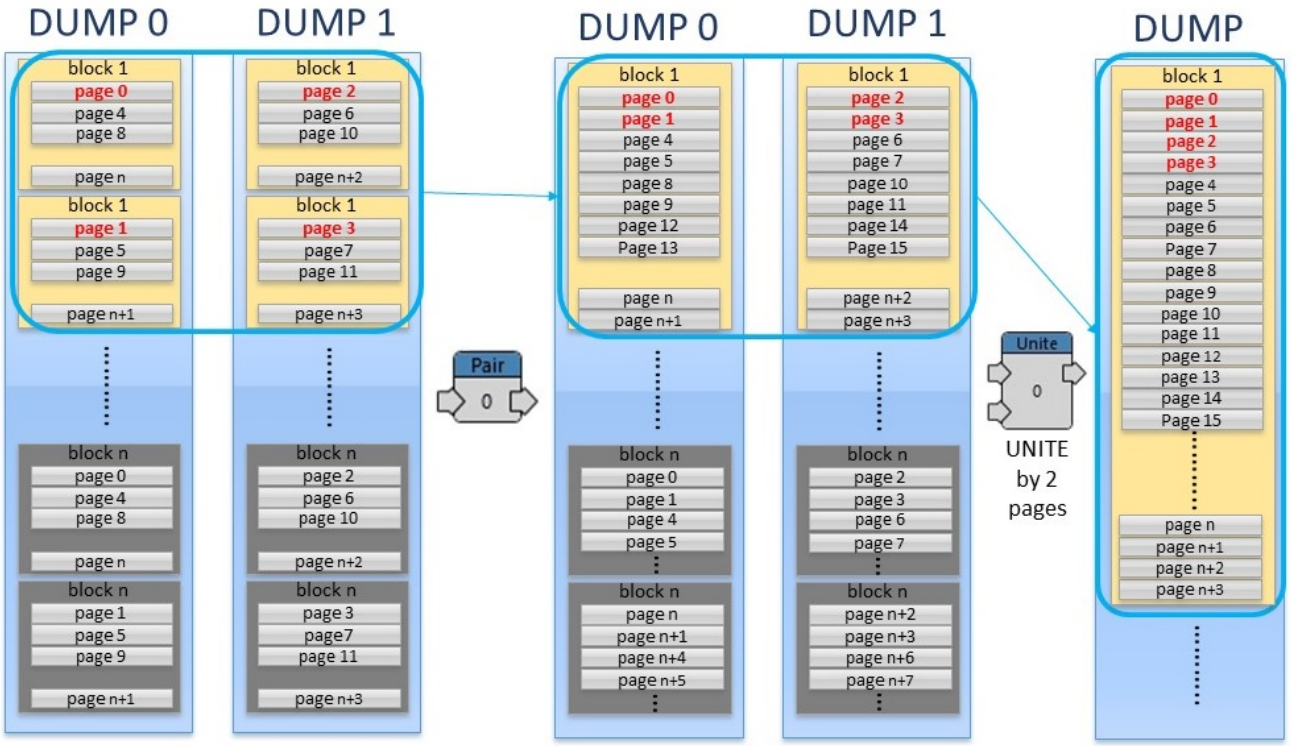
Devices that consist of more than one chip or single chip which has more than 1 crystal might have more complex Page allocation.

The example below shows multi-plane page allocation in case of two chips with two planes each. In this situation, pages are recorded one by one to each plane of each chip.



Therefore in the first dump, physical blocks of the first plane have pages (0,4,8...) and physical blocks of the second plane have pages (1,5,9...). In the second dump (second chip) pages in two neighbour physical blocks will have respectively (2,6,10...) and (3,7,11...).

In such cases to reverse Multiplane page allocation, it's necessary to connect **PAIR** element to each dump at first. By performing this operation pages in the first dump (first chip) will be deployed as following (0,1,4,5,8,9...) and in the second dump, it will be (2,3,6,7,10,11...). So after this transformation, the first dump has two pages go sequentially and the next two pages are located in the second dump. Then to create a dump where ALL pages are located in the correct order, it's necessary to unite two dumps by two pages. This action can be performed with the **UNITE** element.

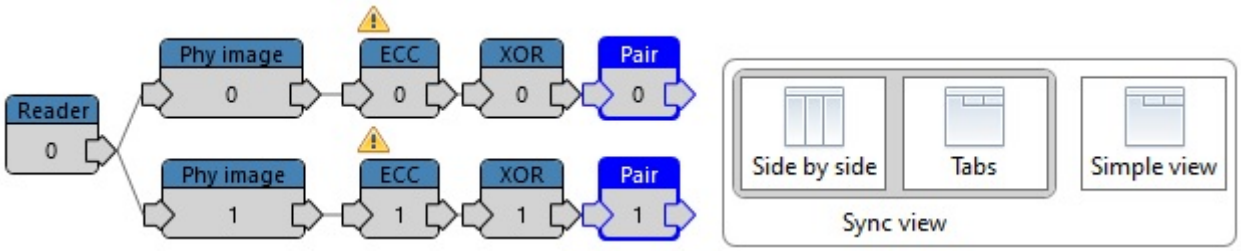


UNITE element has two parameters:

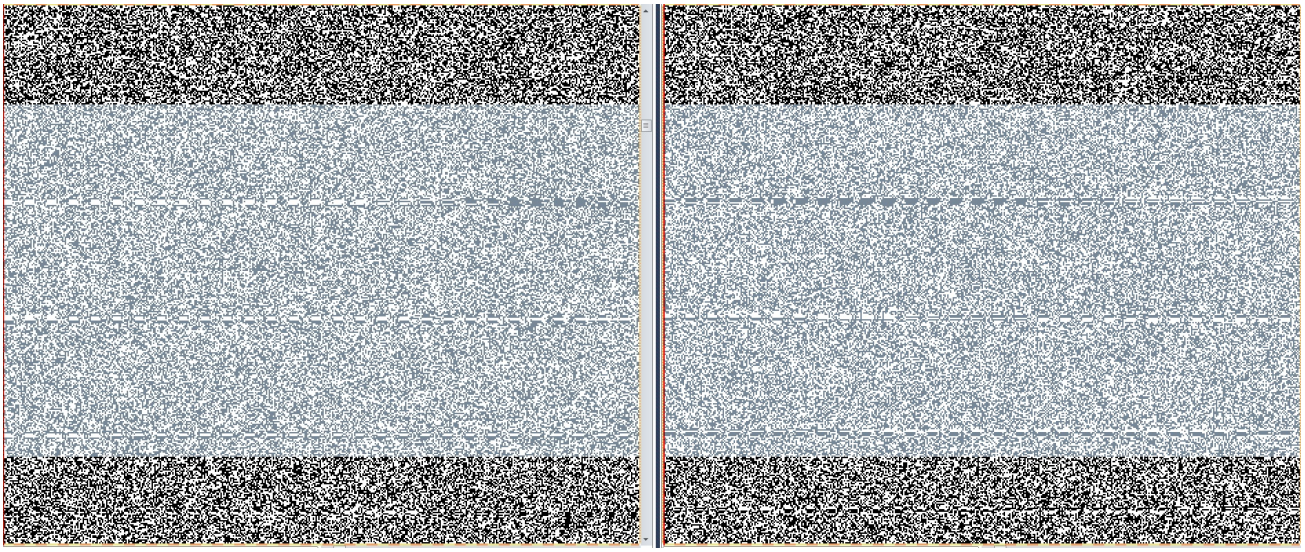
- Number of inputs
- Area size - a value which will be taken from each dump alternately. In this example, it will be 2 pages. (2* page size)

ⓘ If chip consist of 4 planes Unite by 4 pages should be performed.

To detect if Unite by 2 pages should be done, it's necessary to analyze data patterns or LBNs from both dumps by the same offsets.
Navigation group in ' **Side by side'** mode is very helpful here. The same patterns of data by the same offsets are the effect of parallel data writing into both chips.

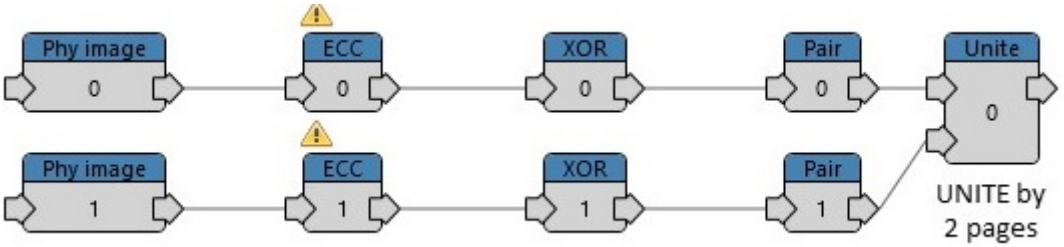


So if similar patterns on the same offsets are found, like on the example below...

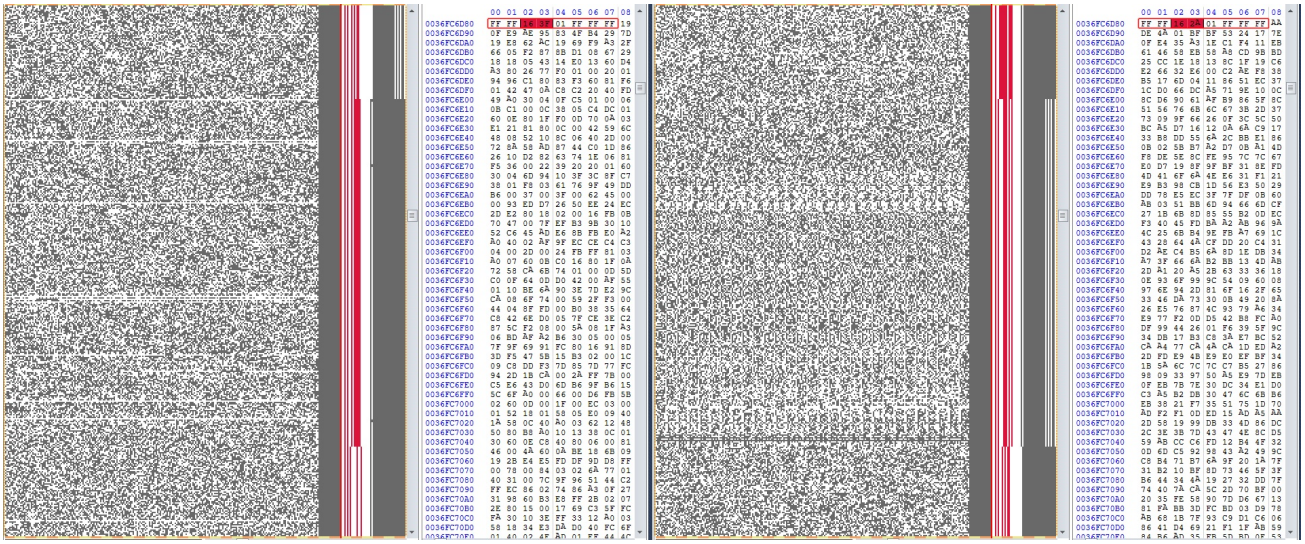


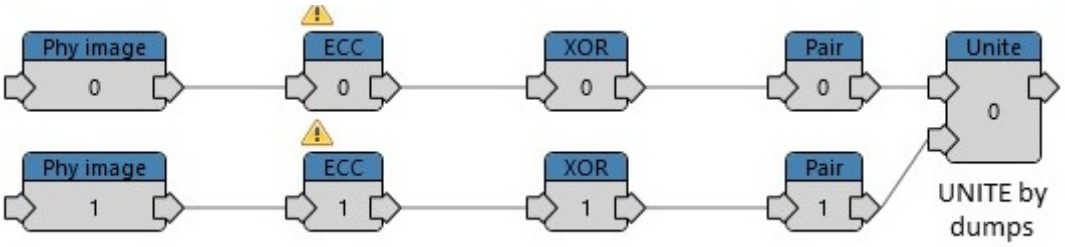
... and LBN's of these blocks are the same.

UNITE by 2 pages is required.

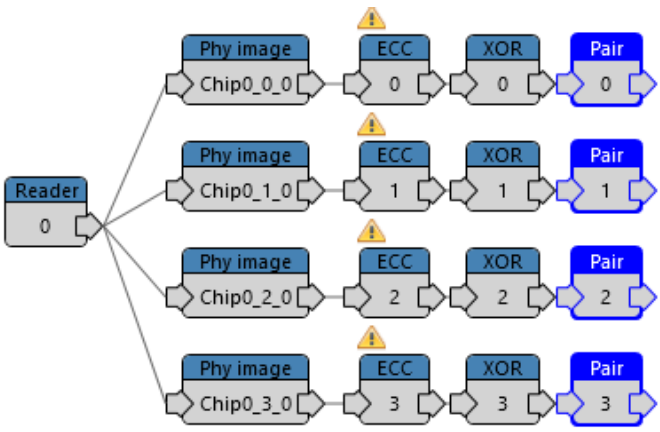


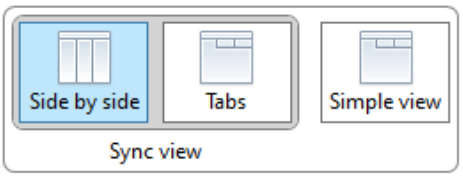
If blocks on the same offsets have a different pattern and LBN's, then data wasn't written simultaneously into the memory chips. Then it is necessary to **UNITE** such dumps by dumps (JBOD like),



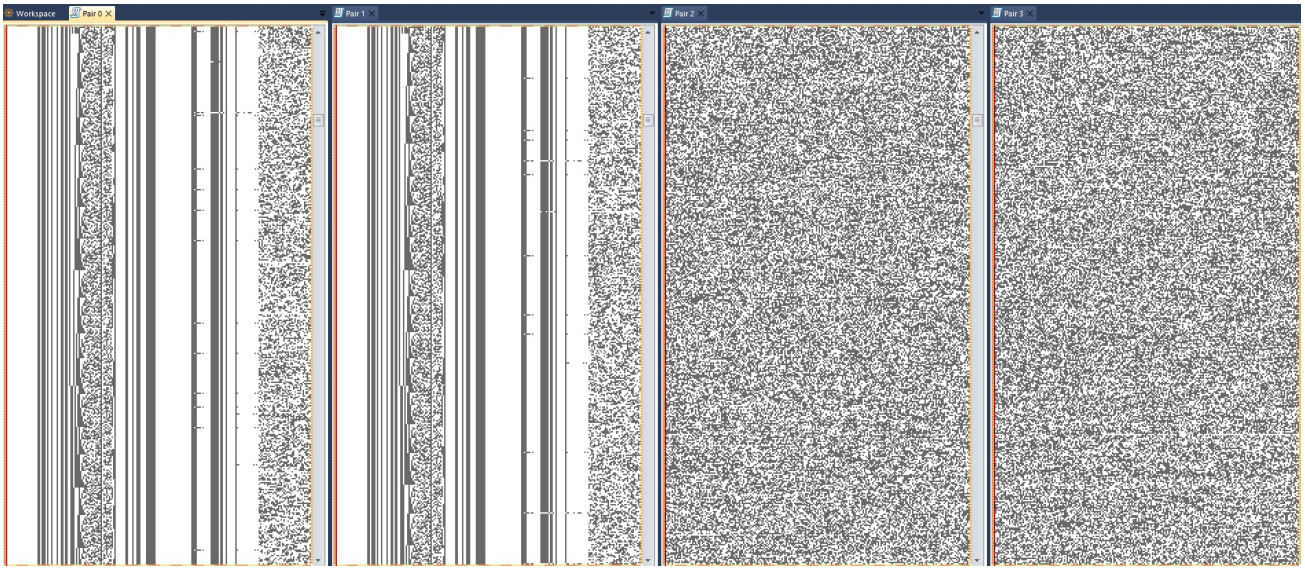


The example below represents the case with 4 chips where data was written simultaneously into every two chips and continuously into those 2 pairs of chips.

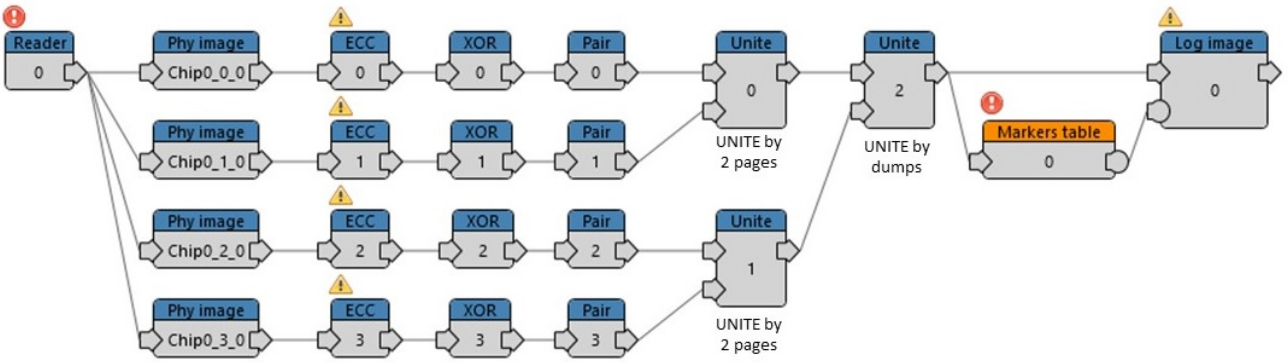




It is visible by similar patterns of data within the first two and second two dumps. At the same time, the patterns of the left and the right pairs of dumps are different.



According to this, we can distinguish that at first, it is necessary to UNITE the first two and the second two dumps by 2 pages. And after that, the results have to be UNITED by dumps.



https://support.rusolut.com/portal/en/kb/articles/multi-plane-page-allocation-10-5-2020